Sourcegraph

3 things to know before building a custom, in-house code search tool

Devs love building tools for devs. It's natural that as your company's codebase grows, devs will start needing code search to understand, fix, and automate across the codebase. Sometimes this means setting up an existing code search tool, but sometimes you might want to build your own code search tool instead. We understand the temptation to build a code search tool—after all, we're building code search at Sourcegraph (it's the foundation of our platform). But here are a few questions you should ask yourself before deciding to build code search inside your company.

Disclaimer: Sourcegraph is a code intelligence platform, and universal code search is a big part of that. So, we are obviously a little biased here.

Will your custom code search tool integrate nicely with your code hosts (GitHub/GitLab/Bitbucket/etc.) and scale to your entire codebase?

A code search tool needs to index all of the repositories from all of your code hosts. That presents several technical challenges:

- Will it scale to the combined size of all of your repositories?
- Will it quickly fetch updates to all branches of all repositories?
- Does it respect the user permissions defined on your code host (so users can only search code they're able to view)?
- Will it put massive load on your code host or get your account banned for abuse?
- If you have code scattered across multiple code hosts, will it support all of them?

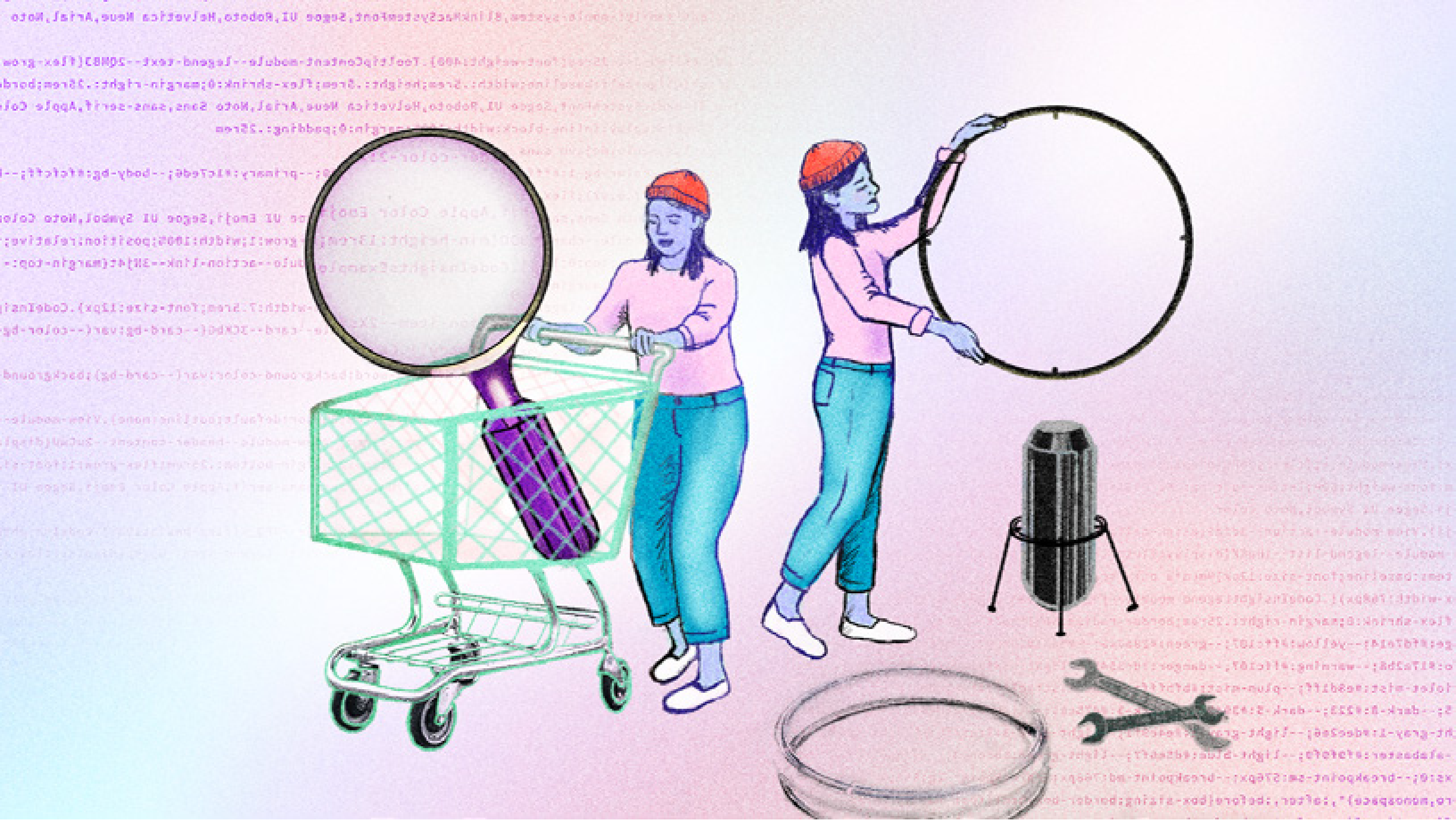
It's a lot of work to handle all of these things cleanly. Here's some proof: as of this writing, there were 858 places dealing with code host API rate limits in Sourcegraph's implementation support for GitHub/GitLab/Bitbucket/etc.

Will your custom code search tool work with all of the programming languages and tools your devs use?

You can't stop at text search when you're building a code search tool. Devs will expect to be able to navigate to definitions and references in code and see documentation at their cursor. All that requires your code search tool to understand language constructs at a syntactic and type level just like a smart IDE.

You can use LSIF to get part of the way here, but that's a lot more work. Just one example: you'll need to be able to adjust LSIF dump files and line ranges forward or backward in commit history so that you can navigate code on commits that don't have a full LSIF dump available.
See Sourcegraph's adjustLocations implementation.

And it's not just programming language support. Devs will want to know other kinds of metadata about the code they're looking at, such as code ownership, test coverage, where it's deployed, what linter issues have been found, etc. The only way to get this kind of information is to ensure your code search tool integrates with all of the tools your developers use every day. And building each one of these integrations is a ton more work.



Who will maintain your custom code search tool, and how much will it cost?

Ever since Jeff Dean and others built the first version of Google's internal code search tool around 2005, the project has cost $100M+ in salaries alone for the dozens of engineers building it (directly or indirectly). You could build one for less money, but it would be much worse. And then why build your own worse code search instead of using one that already exists and is better anyway?

Even if you've built the first version of your own code search tool, you can't stop. Your organization and codebase grows, your preferred programming languages change and evolve, your code hosts change and deprecate APIs, and your devs' needs change in all kinds of unanticipated ways. If you think code search is important enough that it's worth building your own, then you'll surely want to continue investing to keep improving it—but that's expensive.

Finally, focus is crucial. You want your devs solving the problems specific to your business, not problems that are similar across companies. Every company needs code search, for most of the same reasons, and those problems aren't specific to your business.

If you still want to build your own code search tool…

Over the years, we've discovered that companies who need code search also need the ability to understand, fix, and automate changes across their entire codebase, and we've built those capabilities too. If you still want to build your own code search tool, try some existing code search tools first. And then, if you still want to build your own, come talk to us! We've been thinking about code search for almost 10 years by now, and we'd love to share pointers and hear how it's going.

You can get a live demo of Sourcegraph here.

Sourcegraph

about.sourcegraph.com

Sourcegraph is a code intelligence platform that unlocks developer velocity by helping engineering teams understand, fix, and automate across their entire codebase.